Universidade Federal de Viçosa – Campus Florestal

Curso de Ciência da Computação

Disciplina: Banco de Dados

Professor: Daniel Mendes Barbosa

Trabalho 2 — Banco de Dados

Autor:
Erick Vinicius Issa Silva — Matrícula: 05936

Florestal, MG — Novembro de 2025

Sumário

1 Introdução **2**

2 Modelo Relacional Fornecido **2**

3 Diagrama do Banco de Dados **2**

4 Script de Criação do Banco **3**

5 Consultas SQL **6**

5.1 Questão 1 6
5.2 Questão 2 7
5.3 Questão 3 7
5.4 Questão 4 8
5.5 Questão 5 8
5.6 Questão 6 9
5.7 Questão 7 9
5.8 Questão 8 9
5.9 Questão 9 10
5.10 Questão 10 10

6 Expansão do Banco de Dados (Q11) **11**

6.1 Estruturas Novas 11

7 Novos Dados Inseridos (Q12) **12**

8 Novas Perguntas e Consultas (Q13) **14**

9 Conclusão **15**

10 Github **15**

1. Introdução

Este relatório apresenta o desenvolvimento do **Trabalho 2 da disciplina de Banco de Dados**, cujo objetivo é modelar, implementar e consultar um banco de dados relacional baseado em um colecionador de videogames.

Todas as instruções foram realizadas utilizando **MySQL** e **MySQL Workbench**, conforme solicitado. Foram produzidos:

- Diagrama Entidade-Relacionamento (DER);
- Script SQL de criação das tabelas;
- Execução das consultas solicitadas (Questões 1 a 13);
- Propostas de expansão do modelo.

2. Modelo Relacional Fornecido

O esquema relacional a ser implementado é composto pelas seguintes relações:

- `plataforma(idplataforma, nomeplat, ano_lancamento, memoria, idfabricante)`
- `midia(idmidia, nomemid, velocidade_leitura)`
- `compativel_midia(idplataforma, idmidia)`
- `fabricante(idfabricante, nomefab, ano_fundacao)`
- `jogo(idjogo, nomejogo, descricao)`
- `compativel_jogo_plataforma(idplataforma, idjogo, quantidade, ano_lancamento_jogo)`

Atributos `nomeplat`, `nomemid`, `nomefab`, `nomejogo` e `descricao` são do tipo `VARCHAR(30)`. Todos os outros atributos são do tipo `INT`.

3. Diagrama do Banco de Dados

Inclua aqui o PNG exportado do MySQL Workbench:

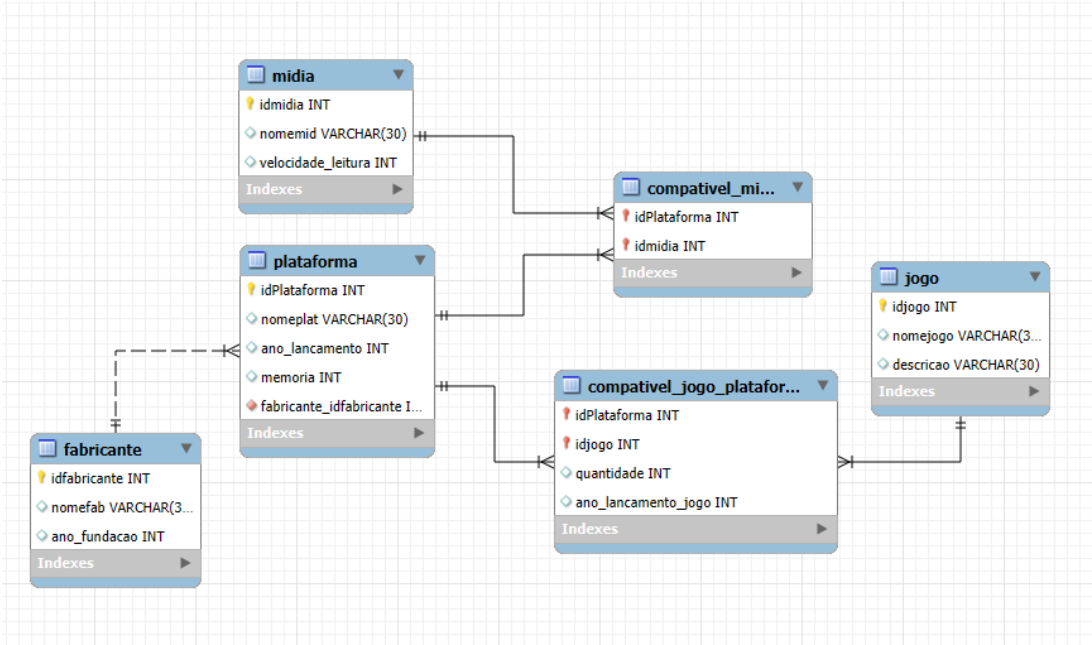

Figura 1: Diagrama ER do banco de dados do colecionador de videogames

4. Script de Criação do Banco

Listing 1: Script de criação do banco de dados

```
-- MySQL Script generated by MySQL Workbench
-- Fri Nov 14 22:10:12 2025
-- Model: New Model    Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
   FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,
   STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,
   ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `trabalhobd` DEFAULT CHARACTER SET
   utf8 ;
```

```sql
USE `trabalhobd` ;

-- -----------------------------------------------------
-- Table `trabalhobd`.`fabricante`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `trabalhobd`.`fabricante` (
  `idfabricante` INT NOT NULL,
  `nomefab` VARCHAR(30) NULL,
  `ano_fundacao` INT NULL,
  PRIMARY KEY (`idfabricante`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `trabalhobd`.`plataforma`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `trabalhobd`.`plataforma` (
  `idPlataforma` INT NOT NULL,
  `nomeplat` VARCHAR(30) NULL,
  `ano_lancamento` INT NULL,
  `memoria` INT NULL,
  `idfabricante` INT NOT NULL,
  PRIMARY KEY (`idPlataforma`),
  INDEX `fk_plataforma_fabricante1_idx` (`idfabricante` ASC)
     VISIBLE,
  CONSTRAINT `fk_plataforma_fabricante1`
    FOREIGN KEY (`idfabricante`)
    REFERENCES `trabalhobd`.`fabricante` (`idfabricante`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `trabalhobd`.`midia`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `trabalhobd`.`midia` (
  `idmidia` INT NOT NULL,
  `nomemid` VARCHAR(30) NULL,
  `velocidade_leitura` INT NULL,
  PRIMARY KEY (`idmidia`))
ENGINE = InnoDB;
```

```sql
-- -----------------------------------------------------
-- Table `trabalhobd`.`jogo`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `trabalhobd`.`jogo` (
  `idjogo` INT NOT NULL,
  `nomejogo` VARCHAR(30) NULL,
  `descricao` VARCHAR(30) NULL,
  PRIMARY KEY (`idjogo`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `trabalhobd`.`compativel_midia`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `trabalhobd`.`compativel_midia` (
  `idPlataforma` INT NOT NULL,
  `idmidia` INT NOT NULL,
  PRIMARY KEY (`idPlataforma`, `idmidia`),
  INDEX `fk_Plataforma_has_midia_midia1_idx` (`idmidia` ASC)
     VISIBLE,
  INDEX `fk_Plataforma_has_midia_Plataforma_idx` (`idPlataforma`
     ASC) VISIBLE,
  CONSTRAINT `fk_Plataforma_has_midia_Plataforma`
    FOREIGN KEY (`idPlataforma`)
    REFERENCES `trabalhobd`.`plataforma` (`idPlataforma`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Plataforma_has_midia_midia1`
    FOREIGN KEY (`idmidia`)
    REFERENCES `trabalhobd`.`midia` (`idmidia`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `trabalhobd`.`compativel_jogo_plataforma`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `trabalhobd`.`
   compativel_jogo_plataforma` (
  `idPlataforma` INT NOT NULL,
  `idjogo` INT NOT NULL,
```

```
  `quantidade` INT NULL,
  `ano_lancamento_jogo` INT NULL,
  PRIMARY KEY (`idPlataforma`, `idjogo`),
  INDEX `fk_Plataforma_has_jogo_jogo1_idx` (`idjogo` ASC) VISIBLE
     ,
  INDEX `fk_Plataforma_has_jogo_Plataforma1_idx` (`idPlataforma`
     ASC) VISIBLE,
  CONSTRAINT `fk_Plataforma_has_jogo_Plataforma1`
    FOREIGN KEY (`idPlataforma`)
    REFERENCES `trabalhobd`.`plataforma` (`idPlataforma`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Plataforma_has_jogo_jogo1`
    FOREIGN KEY (`idjogo`)
    REFERENCES `trabalhobd`.`jogo` (`idjogo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

5. Consultas SQL

A seguir, cada questão.

5.1. Questão 1

```sql
SELECT DISTINCT p.nomeplat
FROM plataforma AS p
NATURAL JOIN compativel_midia AS cm
NATURAL JOIN midia AS m
WHERE m.velocidade_leitura > 100;
```

Result Grid

nomeplat
Atari 2600
NES
SNES

5.2. Questão 2

```sql
SELECT p.nomeplat, p.ano_lancamento
FROM plataforma AS p
NATURAL JOIN fabricante AS f
WHERE f.ano_fundacao > 1970
```

Result Grid

nomeplat	ano_lancamento
Xbox360	2005
Atari 2600	1977

5.3. Questão 3

```sql
SELECT  m.nomemid, m.velocidade_leitura
FROM midia AS m
WHERE m.velocidade_leitura >= 10 AND m.velocidade_leitura <= 30
ORDER BY m.velocidade_leitura DESC, m.nomemid ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nomemid	velocidade_leitura
DVD	20
CD	10

5.4. Questão 4

```sql
INSERT INTO `trabalhobd`.`plataforma`
(idplataforma, nomeplat, ano_lancamento, memoria, idfabricante)
VALUES (
    7,
    'WiiU',
    2012,
    2147483648,
    (SELECT idfabricante
     FROM fabricante
     WHERE nomefab = 'Nintendo')
);
```

5.5. Questão 5

```sql
SELECT p.nomeplat AS plataforma, f.nomefab AS fabricante, SUM(cjp.quantidade) AS numero_jogos
FROM plataforma p
JOIN fabricante f ON p.idfabricante = f.idfabricante
LEFT JOIN compativel_jogo_plataforma cjp ON p.idPlataforma = cjp.idPlataforma
GROUP BY p.idPlataforma, p.nomeplat, f.nomefab
ORDER BY p.nomeplat ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

plataforma	fabricante	numero_jogos
Atari 2600	Atari	4
NES	Nintendo	2
PlayStation3	Sony	5
SNES	Nintendo	1
Wii	Nintendo	4
WiiU	Nintendo	NULL
Xbox360	Microsoft	4

5.6. Questão 6

```
SELECT j.nomejogo
FROM jogo AS j NATURAL JOIN compativel_jogo_plataforma AS cjp
GROUP BY j.idjogo
HAVING COUNT(cjp.idplataforma) >= 2;

SELECT cjp.ano_lancamento_jogo, COUNT(*) AS jogos_lancados
FROM compativel_jogo_plataforma AS cjp NATURAL JOIN plataforma AS p
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nomejogo
Mario Bros
F-Zero
Super Mario Bros
FIFA12
Street Fighter IV

5.7. Questão 7

```
SELECT cjp.ano_lancamento_jogo, COUNT(*) AS jogos_lancados
FROM compativel_jogo_plataforma AS cjp
GROUP BY cjp.ano_lancamento_jogo
ORDER BY cjp.ano_lancamento_jogo ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ano_lancamento_jogo	jogos_lancados
1982	1
1983	2
1985	1
1990	1
2006	1
2007	2
2008	2
2010	1
2011	4
2012	1

5.8. Questão 8

```sql
SELECT  j.nomejogo,  cjp.ano_lancamento_jogo
FROM compativel_jogo_plataforma AS cjp NATURAL JOIN jogo AS j
ORDER BY cjp.ano_lancamento_jogo ASC
LIMIT 1
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

nomejogo	ano_lancamento_jogo
River Raid	1982

5.9. Questão 9

```sql
SELECT AVG(p.memoria)
FROM  plataforma AS p NATURAL JOIN compativel_jogo_plataforma AS cjp
WHERE cjp.quantidade > 0;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVG(p.memoria)
257958160.0000

5.10. Questão 10

```sql
SELECT f.nomefab
FROM fabricante AS f NATURAL JOIN plataforma AS p NATURAL JOIN midia AS m
GROUP BY idfabricante
HAVING COUNT(DISTINCT m.idmidia) > 1
ORDER BY f.nomefab;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nomefab
Atari
Microsoft
Nintendo
Sony

6. Expansão do Banco de Dados (Q11)

- A tabela fabricante_midia, relaciona os fabricantes de plataformas também a mídias, com a adição do campo ano_lancamento_midia, é importante ressaltar que nesse contexto, as mídias são fabricadas somente por fabricantes que também fazem plataformas;
- A tabela controle é relacionada a plataforma e a uma fabricante, possuindo tembém os atributos nomecontrole e descricao_controle;

6.1. Estruturas Novas

Listing 2: Novas estruturas adicionadas

```
CREATE TABLE fabricante_midia (
    idmidia INT NOT NULL,
    idfabricante INT NOT NULL,

    PRIMARY KEY (idmidia, idfabricante),


    FOREIGN KEY (idmidia)
        REFERENCES midia(idmidia)
        ON DELETE CASCADE
        ON UPDATE CASCADE,

    FOREIGN KEY (idfabricante)
        REFERENCES fabricante(idfabricante)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

ALTER TABLE fabricante_midia
ADD COLUMN ano_lancamento_midia INT;

CREATE TABLE controle (
    idcontrole INT NOT NULL,
    idplataforma INT NOT NULL,
    idfabricante INT NOT NULL,
    nomecontrole VARCHAR(35),
    descricao_controle VARCHAR(35) NULL,

    PRIMARY KEY (idcontrole),
```

```
    FOREIGN KEY (idplataforma)
        REFERENCES plataforma(idplataforma)
        ON DELETE CASCADE
        ON UPDATE CASCADE,

    FOREIGN KEY (idfabricante)
        REFERENCES fabricante(idfabricante)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

Inclua o novo diagrama:

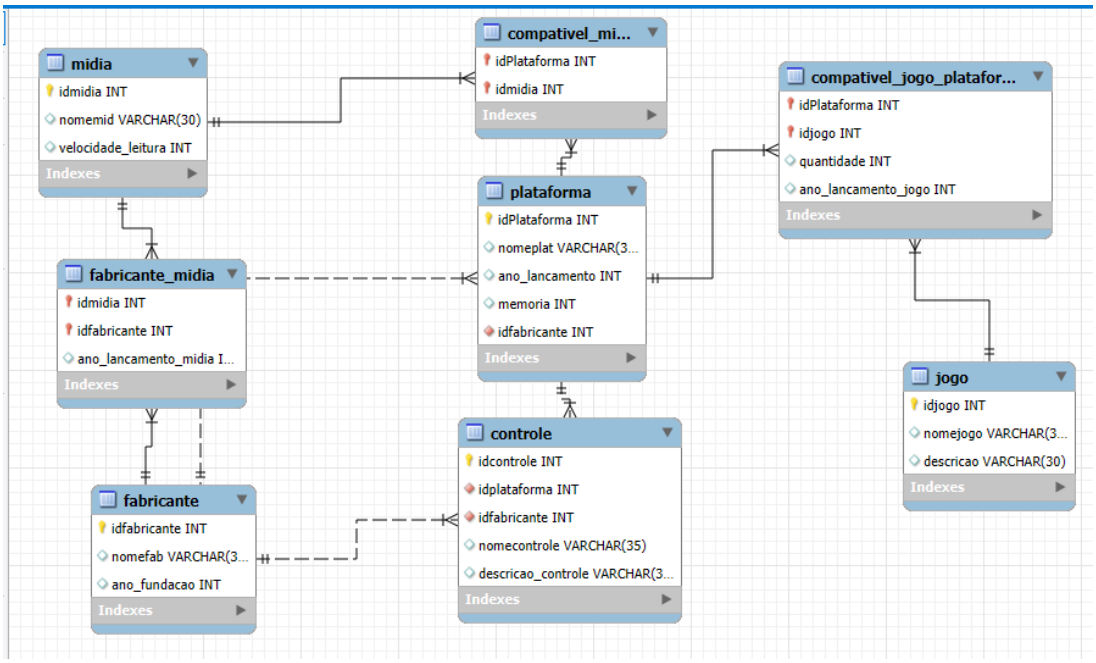

Figura 2: Diagrama ER expandido

7. Novos Dados Inseridos (Q12)

Listing 3: Comandos INSERT/UPDATE

```
INSERT INTO fabricante_midia (idmidia, idfabricante,
   ano_lancamento_midia) VALUES
(1, 1, 2020),
(1, 3, 2021),

(2, 1, 2020),
(2, 2, 2023),
(2, 4, 2024),
```

```sql
(3, 3, 2019),

(4, 2, 2022),
(4, 4, 2024),

(5, 1, 2025),

(6, 3, 2021),

(7, 4, 2023);

INSERT INTO controle (idcontrole, idplataforma, idfabricante,
   nomecontrole) VALUES
(1, 2, 1, 'NES controller'),
(2, 3, 1, 'SNES controller'),
(3, 4, 1, 'Wii Remote'),
(4, 6, 2, 'Dualshock 3'),
(5, 5, 3, 'Xbox Wireless Controller'),
(6, 1, 4, 'Atari 2600 controller');

UPDATE controle
SET descricao_controle = 'D-pad, A, B, Start, Select'
WHERE idcontrole = 1;

UPDATE controle
SET descricao_controle = 'Adiciona X, Y e bot es L/R (ombro)'
WHERE idcontrole = 2;

UPDATE controle
SET descricao_controle = 'Sensor de movimento e ponteiro IR'
WHERE idcontrole = 3;

UPDATE controle
SET descricao_controle = 'Vibra  o (rumble) e anal gicos'
WHERE idcontrole = 4;

UPDATE controle
SET descricao_controle = 'Padr o do 360, bot es A,B,X,Y'
WHERE idcontrole = 5;

UPDATE controle
SET descricao_controle = 'Joystick cl ssico com um bot o'
```

```
WHERE idcontrole = 6;
```

8. Novas Perguntas e Consultas (Q13)

Segue as perguntas:

- Liste o nome, plataforma e fabricante dos controller que possuem "controller"em seu nome, considere case insensitive.

```
351     -- Q13 -- Liste o nome, plataforma e fabricante dos controller que possuem "controller"
352     -- em seu nome, considere case insensitive
353 •   SELECT  c.nomecontrole AS nome_do_controle, p.nomeplat AS plataforma, f.nomefab AS fabricante,
354     c.descricao_controle AS descricao
355     FROM controle AS c NATURAL JOIN plataforma AS p
356     NATURAL JOIN  fabricante AS f
357     WHERE c.nomecontrole LIKE '%controller%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nome_do_controle	plataforma	fabricante	descricao
NES controller	NES	Nintendo	D-pad, A, B, Start, Select
SNES controller	SNES	Nintendo	Adiciona X, Y e botões L/R (ombro)
Xbox Wireless Controller	Xbox360	Microsoft	Padrão do 360, botões A,B,X,Y
Atari 2600 controller	Atari 2600	Atari	Joystick clássico com um botão

Figura 3: Questão 13 - 1ª criada

- Liste o nome da fabricante e o nome da mídia na qual a plataforma e a midia possuem o mesmo fabricante entre si.

```
58
59      -- Q13(2) -- Liste o nome da fabricante e o nome da mídia na qual a plataforma e a midia possuem
60      -- o mesmo fabricante entre si
61 •    SELECT DISTINCT f_plat.nomefab AS fabricante, m.nomemid AS nome_da_midia
62      FROM  fabricante AS f_plat NATURAL JOIN plataforma AS p
63      NATURAL JOIN  compativel_midia AS cm
64      NATURAL JOIN  midia AS m
65      NATURAL JOIN fabricante_midia AS fm
66      NATURAL JOIN fabricante AS f_midia
67      WHERE f_plat.nomefab = f_midia.nomefab;
68
69
70
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

fabricante	nome_da_midia
Nintendo	DVD
Nintendo	SD

Figura 4: Questão 13 - 2ª criada

9. Conclusão

Neste trabalho foi possível exercitar:

- Modelagem relacional;
- Criação de bancos no MySQL;
- Consultas SQL de diferentes níveis de complexidade;
- Expansão do modelo relacional;
- Análise dos resultados.

10. Github

Segue o link do repositório Github:
https://github.com/ErickIssa/BancoDeDados—TP2